פרק ראשון - משלוחי פיצה (90 נק')

חלק א'- מבוא והנחיות (3 נק')

תרגיל 1-

Deliveries k	Problem without fuel	Problem with fuel $(l = 5)$
1	1	1
2	2	10
3	6	150
4	24	3000
5	120	75000
6	720	2250000
7	5040	78750000
8	40320	3150000000
9	362880	141750000000
10	3628800	7087500000000

חלק ג'- הגדרת מרחבי החיפוש של מרחבי מסלולי נסיעת הטוסטוס (15 נק')

תרגיל 2-

מקדם הסיעוף המינימלי הינו 0 כאשר במצה ההתחלתי אין מספיק דלק לעבור לאף מיקום אחר במפה ולכן בפרט לא קיימים עוד מצבים כי שני האופרטורים מחזירים קבוצות ריקות.

מקדם הסיעוף המקסימלי הינו $k+l$ כאשר ניתן להגיע מכל מיקום של הזמנה אל כל תחנת דלק או מיקום אחר של הזמנה.

תרגיל 3-

כן, יכול להיות מעגלים בגרף בין תחנות דלק לדוגמא: שלומי נמצא בתחנת דלק f_j והופעל אופרטור עבור תדלוק, משם שלומי עובר לתחנת דלק $f_i \mid i \neq j$ ומשם שוב מופעל אופרטור תדלוק ונחזור לתחנת דלק f_j (נשים לב שאופרטור זה לא משנה את סטטוס ההזמנות).

תרגיל 4-

לכאורה יכולים להיות ∞ מצבים מכיוון ש d הינו מספר ממשי ולכן מתנהג ברציפות. אמנם, אצלנו בבעיה d קטן רק לפי מרחק נתון בין המיקומים על המפה ולכן לא כולם יהיו ישיגים. לדוגמא אם כל המרחקים בין שני מיקומים הם שלמים לעולם לא נגיע למיקום עם d לא שלם ולכן בהכרח לא נבקר בכל המצבים האפשריים.

תרגיל 5-

כן, ייתכנו בורות ישיגים מהמצב ההתחלתי שאינם מצבי מטרה לדוגמא:
לשלומי יש יותר מהזמנה אחת והמרחק בין כל שני צמתים על המפה מקיים $Dist(v_1,v_2)>0 \mid v_1 \neq v_2 \wedge v_1,v_2 \in V$ ובמצב ההתחלתי הופעל אופרטור עבור הורדת הזמנה אשר הביא את שלומי למצב הבא: $(t_i,0,T\setminus\{i\},\{i\}) \mid i \in T$ כלומר שלומי הצליח למסור את **אחת** ההזמנות אבל נגמר לו הדלק ולכן הפעלת כל אחד מהאופרטורים על המצב הנ"ל יניב קבוצה ריקה ולכן הגענו לבור כי אין ממנו קשתות יוצאות.

תרגיל 6-

הגדרה פורמאלית לפונקציית העוקב:

$$Succ : S \to P(S)$$
$$Succ(v_1,d_1,T_1,F_1) =$$
$$\left\{(v_2,d_2,T_2,F_2) \mid d_2 = d_{refuel}, T_2 = T_1, F_2 = F_1, v_2 \in GasStations \wedge d_1 \geq Dist(v_1,v_2)\right\}$$
$$\bigcup\left\{(v_2,d_2,T_2,F_2) \mid d_1 \geq Dist(v_1,v_2) \wedge \left(\exists t_{i\in[k]} \in Ord \mid t_i = v_2 \wedge i \in T_1\right), d_2 = d_1 - Dist(v_1,v_2), T_2 = T_1 \setminus \{i\}, F_2 = F_1 \bigcup \{i\}\right\}$$

נכתב על ידי: עידו יחזקאל ואוהד זוהר

<u>תרגיל 7-</u>

תחת הנחה זו החסם התחתון של העומק המינימלי של מצב מטרה הינו בדיוק k זאת מכיוון שנצטרך לעבור בכל אחת מ k ההזמנות **השונות** כלומר ב- k **מצבים שונים** לכל הפחות. מצב זה ייתכן כאשר במצב ההתחלתי לשלומי יש מספיק דלק בקטנוע לבצע מסלול זה ללא צורך לתדלק.

חלק ד'- מתחילים לתכנת (7 נק')

תרגיל 8-

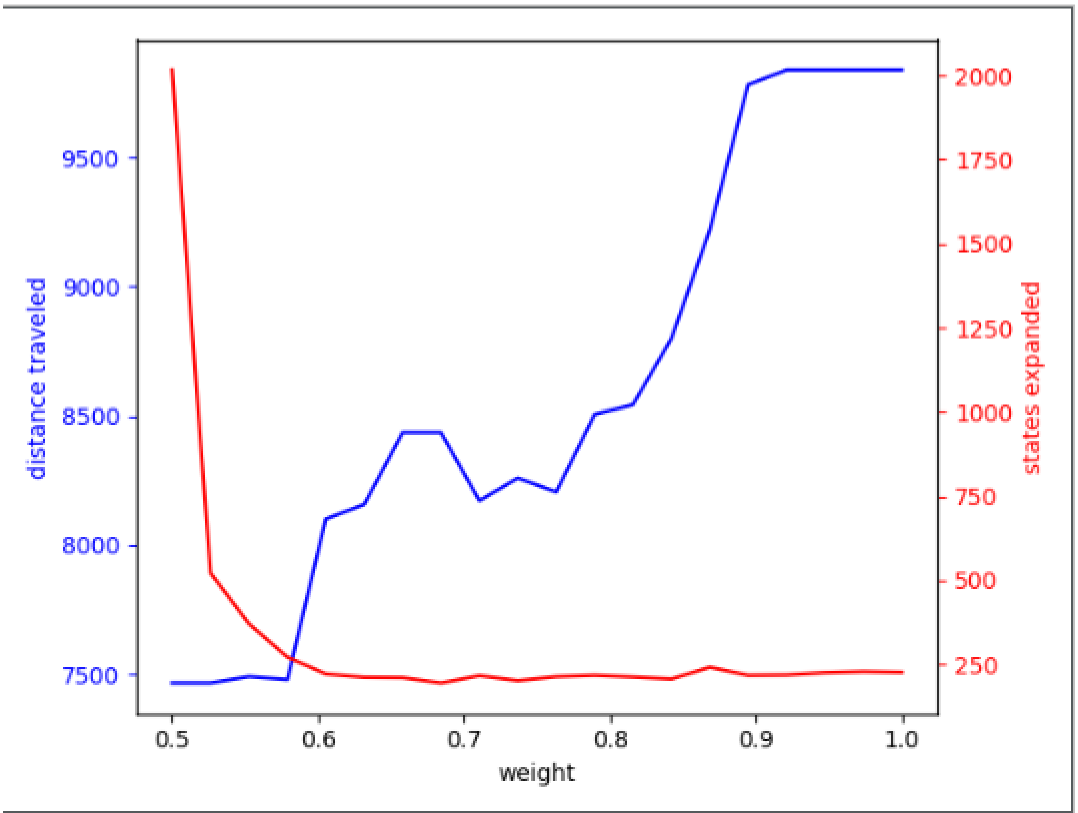
```
Map(src: 54 dst: 549)

UniformCost

time:   0.51

#dev: 17355

total_cost:  7465.52897
|path|: 137

path: [54, 55, 56, 57, 58, 59, 60, 28893, 14580, 14590, 14591, 14592,
14593, 81892, 25814, 81, 26236, 26234, 1188, 33068, 33069, 33070,
15474, 33071, 5020, 21699, 33072, 33073, 33074, 16203, 9847, 9848,
9849, 9850, 9851, 335,  9852, 82906, 82907, 82908, 82909, 95454,
96539, 72369, 94627, 38553, 72367, 29007,94632, 96540, 9269, 82890,
29049, 29026, 82682, 71897, 83380, 96541, 82904, 96542, 96543, 96544,
96545, 96546,96547, 82911, 82928, 24841, 24842, 24843, 5215, 24844,
9274, 24845, 24846, 24847, 24848, 24849, 24850, 24851, 24852, 24853,
24854, 24855, 24856, 24857, 24858, 24859, 24860, 24861, 24862, 24863,
24864, 24865, 24866, 82208, 82209, 82210, 21518, 21431, 21432, 21433,
21434, 21435, 21436, 21437, 21438, 21439, 21440, 21441, 21442, 21443,
21444, 21445, 21446, 21447, 21448, 21449, 21450, 21451, 621, 21452,
21453, 21454, 21495, 21496, 539, 540,541, 542, 543, 544, 545, 546,
547, 548, 549]
```

תרגיל 11-

```
Map(src: 54 dst: 549)
A* (h=AirDist, w=0.500)
time:   0.07
#dev: 2016
total_cost:  7465.52897
|path|: 137
path: [54, 55, 56, 57, 58, 59, 60, 28893, 14580, 14590, 14591, 14592,
14593, 81892, 25814, 81, 26236, 26234,
  1188, 33068, 33069, 33070, 15474, 33071, 5020, 21699, 33072, 33073,
33074, 16203, 9847, 9848, 9849, 9850,
  9851, 335, 9852, 82906, 82907, 82908, 82909, 95454, 96539, 72369,
94627, 38553, 72367, 29007, 94632, 96540,
  9269, 82890, 29049, 29026, 82682, 71897, 83380, 96541, 82904,
96542, 96543, 96544, 96545, 96546, 96547, 82911,
  82928, 24841, 24842, 24843, 5215, 24844, 9274, 24845, 24846, 24847,
24848, 24849, 24850, 24851, 24852, 24853,
  24854, 24855, 24856, 24857, 24858, 24859, 24860, 24861, 24862,
24863, 24864, 24865, 24866, 82208, 82209, 82210,
  21518, 21431, 21432, 21433, 21434, 21435, 21436, 21437, 21438,
21439, 21440, 21441, 21442, 21443, 21444, 21445,
  21446, 21447, 21448, 21449, 21450, 21451, 621, 21452, 21453, 21454,
21495, 21496, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549]
```

חלק ה'- אלגוריתם A^* (10 נק')

תרגיל 12- A^*State Expanded and Solution Distance Vs Heuristic Weight



הסבר הגרף:

כפי שראינו בהרצאה ככל שאנו מגדילים את משקל הפונקציה היוריסטית על פני פונקציית המרחק האלגוריתם A^* קורס ל Greedy Best 1st וכך הפתרון שאנו מקבלים מתרחק מהפתרון האופטימלי אך אנחנו מקבלים ביצועים טובים יותר כי האלגוריתם מפתח פחות צמתים.

לעומת זאת ככל שפונקציית המרחק והפונקציה היוריסטית ממושקלות באופן שווה כן מובטח לנו פתרון אופטימלי אך מספר הצמתים שהאלגוריתם מפותח גדל.

נכתב על ידי: עידו יחזקאל ואוהד זוהר

חלק ו'- בעיית המשלוחים המופשטת (10 נק')

תרגיל 14-

היוריסטיקה הנתונה בן קבילה, בהינתן מצב s על מנת להגיע ממנו למצב מטרה נצטרך לסיים את כל ההזמנות השייכות ל $s.T$ בפרט את **ההזמנה הכי רחוקה** ממנו כלומר המסלול למצב מטרה בעל מרחק שהוא **לפחות** המרחק מהמצב s להזמנה הכי רחוקה (בדומה לבעיית הפאזל והיוריסטיקת מנהטן אשר צריך להזיז את המשבצת לפחות הערך של היוריסטיקה מנהטן).

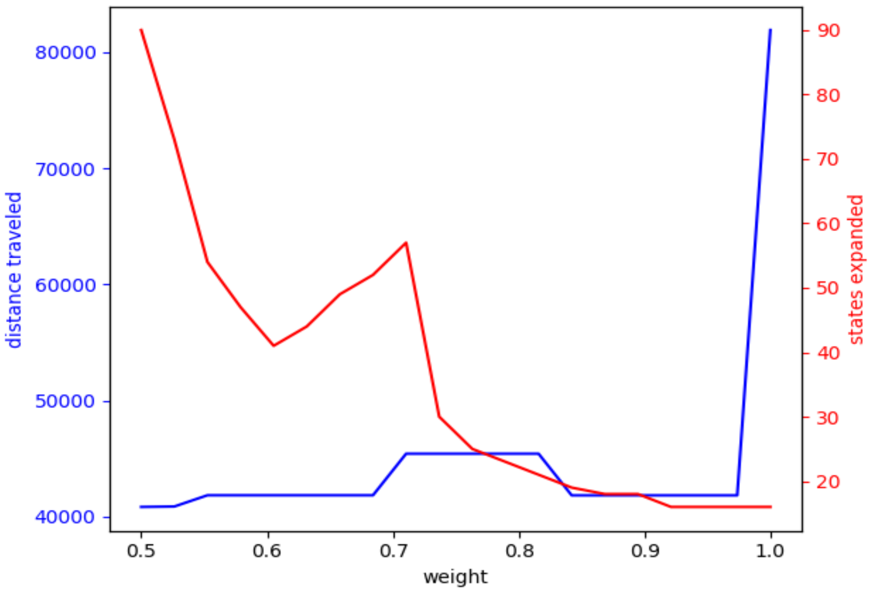
תרגיל 16-

```
RelaxedDeliveries(big_delivery)
A* (h=MaxAirDist, w=0.500)
time:   3.13
#dev: 6836
total_cost: 40844.21165
|path|: 11
path: [33919, 18409, 77726, 26690, 31221, 63050, 84034, 60664, 70557,
94941, 31008]
gas-stations: [31221, 70557]
```

תרגיל 17-

```
RelaxedDeliveries(big_delivery)
A* (h=MSTAirDist, w=0.500)
time:   0.97
#dev: 90
total_cost: 40844.21165
|path|: 11
path: [33919, 18409, 77726, 26690, 31221, 63050, 84034, 60664, 70557,
94941, 31008]
gas-stations: [31221, 70557]
```

תרגיל 18-



נכתב על ידי: עידו יחזקאל ואוהד זוהר

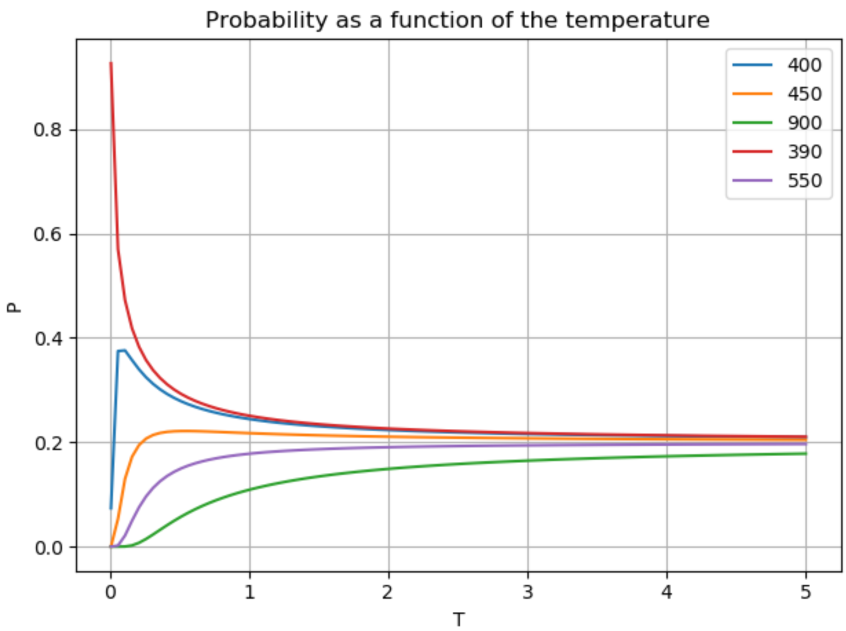
<u>חלק ז'- אלגוריתם חיפוש חמדני-סטוכאסטי (20 נק')</u>

<u>תרגיל 19- הוכחה כי שינוי הסקאלה אינו משפיע על פונקציית התפלגות</u>

$$\forall x_i \in x^t : \Pr(x_i) = \frac{\left(\frac{x_i}{\alpha}\right)^{-\frac{1}{T}}}{\sum_{pnt_h \in best\ N} \left(\frac{x_h}{\alpha}\right)^{-\frac{1}{T}}} =$$

$$\frac{\frac{x_i^{-\frac{1}{T}}}{\alpha^{-\frac{1}{T}}}}{\sum_{pnt_h \in best\ N} \frac{x_i^{-\frac{1}{T}}}{\alpha^{-\frac{1}{T}}}} = \frac{\frac{1}{\alpha^{-\frac{1}{T}}} \cdot x_i^{-\frac{1}{T}}}{\frac{1}{\alpha^{-\frac{1}{T}}} \cdot \sum_{pnt_h \in best\ N} x_i^{-\frac{1}{T}}} = \frac{(x_i)^{-\frac{1}{T}}}{\sum_{pnt_h \in best\ N} (x_i)^{-\frac{1}{T}}}$$

<u>תרגיל 20-</u>



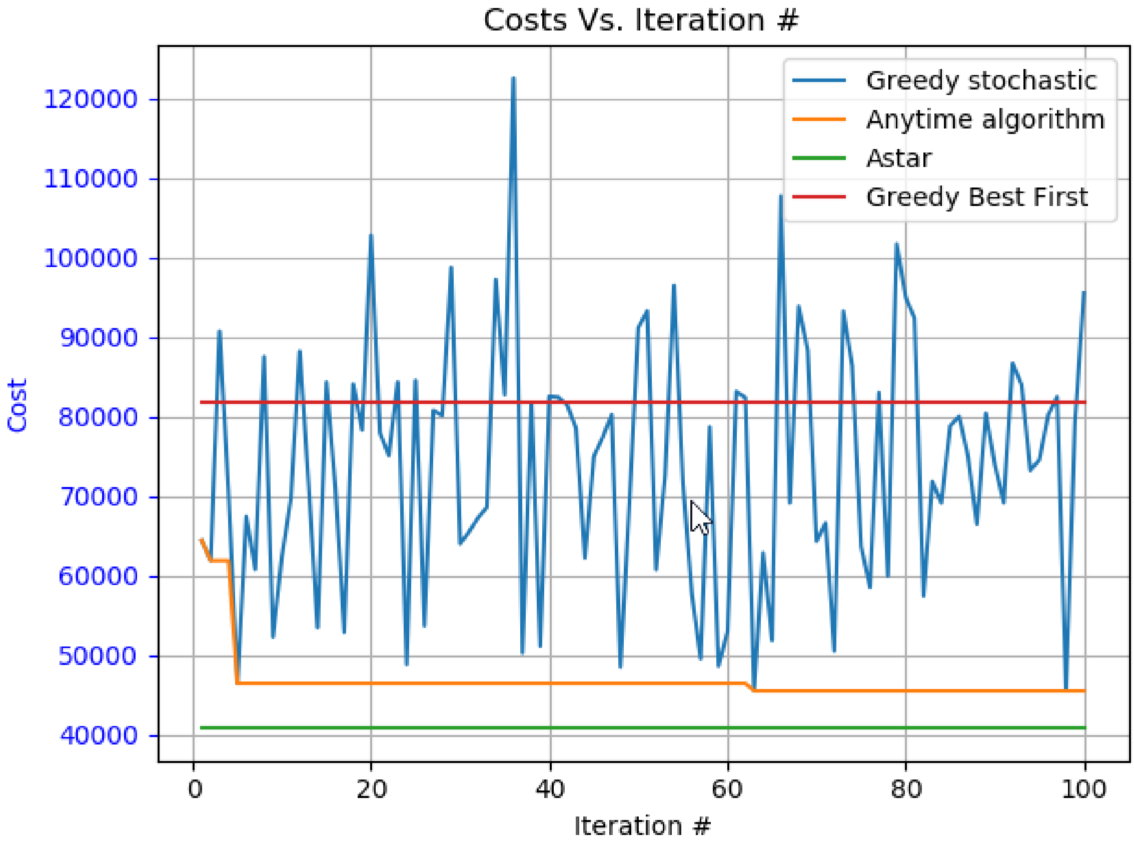
<u>תרגיל 21-</u>

עבור $T \to 0$ האלגוריתם "ילך על בטוח" ויבחר במצב עם הערך היוריסטי הכי נמוך וזאת גם על ידי הסתכלות על הגרף וגם על ידי ניתוח נומרי.

<u>תרגיל 22-</u>

עבור $T \to \infty$ פונקציית ההסתברות הופכת להיות בעלת הסתברות אחידה (במקרה שלנו עבור $N = 5$ ערך פונקציית ההסתברות יהיה $\frac{1}{5} = 0.2$ כפי שרואים בגרף) לכן לכל אחד מהמצבים יש הסתברות שווה להיבחר.

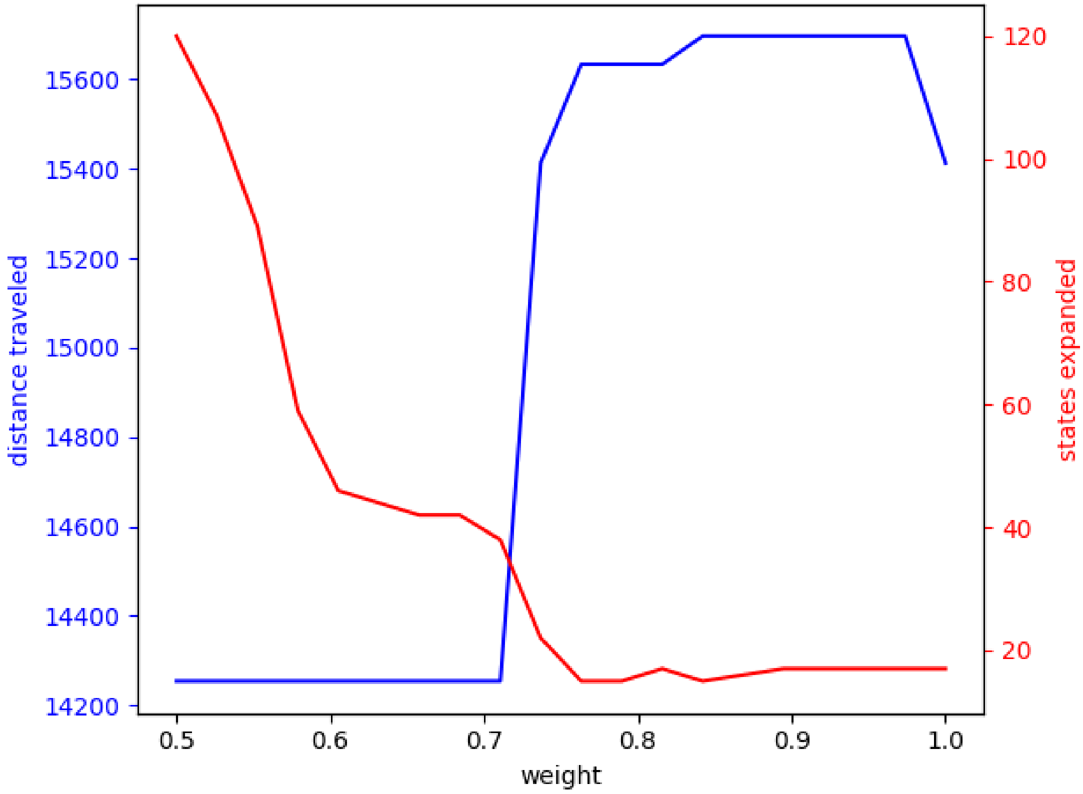
תרגיל 24-



נכתב על ידי: עידו יחזקאל ואוהד זוהר

חלק ח'- אלגוריתם מבוסס A^* (20 נק')

תרגיל 26-



תרגיל 27-

היוריסטיקה בה נשתמש היא :

$h(s) = \{\text{cost from s to target state when s is init state in the relax delivery problem}$

$^*\text{if there is no solution return } \infty$

הוכחה כי היוריסטיקה קבילה:

נסמן ב - $h^*(s)$ את היוריסטיקה המושלמת. יהי s מצב כלשהו, צריך להוכיח כי $\forall s: 0 \le h(s) \le h^*(s)$ בנוסף כאשר S מצב מטרה מתקיים $h(s)=0$.

- עבור S מצב מטרה בפרט העלות בבעיה המופשטת יותר ממנו למצב מטרה היא אפס ולכן מתקיים: $h(s)=0$.
- נשים לב כי בשתי הגרסאות לבעיה המחיר בין שני מצבים הוא המרחק בין הצמתים (מיקום גאוגרפי) של המצבים, כאשר בבעיה המופשטת המחיר נקבע על ידי **המרחק האווירי**, כלומר יוריסטיקה זו מחזירה לנו את מחיר **המסלול האופטימלי** בין מצב למצב מטרה כאשר המחיר נקבע על ידי מרחק אווירי בין כל שני מצבים במסלול נקבל על ידי המרחק האווירי בין הצמתים שלהם.

בפרט בבעיית ה- Strict Delivery נצטרך לעבור בלפחות אותם מצבים במסלול (על מנת לסיים את ההזמנות וכולל עצירות בתחנות דלק, אולי גם

נוספות עצירות בתחנת דלק כי כעת המרחק בין שני צמתים גדל), ובפרט המחיר בין שני צמתים בבעיה מופשטת קטן שווה מהמחיר בין שני צמתים על המפה הרי שבפרט מחיר המסלול שהבעיה המופשטת מחזירה **קטן שווה** מהמחיר של המסלול האופטימלי בבעיית Strict Delivery.
ולכן היוריסטיקה קבילה.

תרגיל 28-

Relaxed Deliveries Heuristic output:

```
StrictDeliveries(small_delivery)
A* (h=RelaxedProb, w=0.500)
time:  4.66
#dev: 80
total_cost: 14254.79234
|path|: 8
path: [43516, 67260, 17719, 43454, 43217, 32863, 7873, 42607]
gas-stations: [17719, 32863]
```

MST Air Dist Heuristic output:

```
StrictDeliveries(small_delivery)
A* (h=MSTAirDist, w=0.500)
time:   8.06
#dev: 120
total_cost: 14254.79234
|path|: 8
path: [43516, 67260, 17719, 43454, 43217, 32863, 7873, 42607]
gas-stations: [17719, 32863]
```

כפי שניתן לראות היוריסטיקה הניבה ביצועים טובים יותר, **גם מספר פיתוח המצבים ירד וגם זמן הריצה השתפר.** תוצאה זו אפשרית מכיוון שיוריסטיקה זו יותר מיודעת ולכן מניבה תוצאות טובות יותר, ניתן להסביר זאת על ידי שהיוריסטיקה המשתמשת בבעיה המופשטת מחשבת מחיר מסלול אופטימלי בין מצבים ומתחשבת בעוד פרמטרים אשר היוריסטיקה הקודמת לא התחשבה בהם כמו: ההזמנות שבוצעו, מצב הדלק של המצב.

פרק שני – שאלה תאורטית (10 נק')

א. תהי h המתוארת בשאלה קבילה צריך להוכיח h_0 גם קבילה, הוכחה:

יהי **s** מצב כלשהו, נפריד למקרים:

- <u>מקרה א:</u> h מוגדרת עבור ולכן מקבילות של h והגדרת h_0 בפרט

  $h_0(s) = h(s) \Rightarrow h_0(s) \le h^*(s)$
- <u>מקרה ב:</u> h לא מוגדרת עבורו ולכן $h_0(s) = 0$ ומכיוון שמתקיים

  $0 \le h^*(s)$ נקבל $h_0(s) \le h^*(s)$.

כלומר $0 \le h_0(s) \le h^*(s)$ לכל **s** כלשהו בבעיה.

ב. יוריסטיקה קבילה ומיודעת יותר מ h_0: נסמנה ב h_1.

- עבור מצבים בהם h מוגדרת נגדיר: $h_1(s) = h(s)$.
- עבור מצבים עליהם h לא מוגדרת $\left(s \in S^{'}\right)$ נבצע את האלגוריתם הבא לחישוב $h_1(s)$:

```
def h1_value(State s):
	if is_goal(s)
		return 0
	h1_v = 0
	for curr_succ in succ(s):
		if ApplicableH(curr_succ):
			h_v_succ = h(curr_succ)
			if h1_v == 0 or h_v_succ < h1_v:
				h1_v = h_v_succ
	return h1_v
```

- נשיב לי כי האלגוריתם עומד בתנאי הסיבוכיות כי אנו מבצעים לולאות בעלת סיבוכיות של $O(b)$.
- בנוסף הפונקציה שהגדרנו מיודעת יותר מ h_0 (לפי ההגדרה בשאלה) ובנוסף גם קבילה זאת מכיוון שעבור מצבים ש h מוגדרת עליהם היא שווה לערך של h_0 ולכן מיודעת וקבילה עבור מצבים אלו.

עבור מצבים ש h לא מוגדרת עליהם הערך שהיא מחזירה גדול שווה מ-0 ולכן גדול שווה מ h_0 בנוסף הערך היוריסטי של מצב כזה בהכרח קטן שווה לערך של היוריסטיקה המושלמת, מכיוון שאם ערכו הוא 0 אז הטענה מתקיימת, אחרת, גדול ממש מ-0 לכן שווה לערך היוריסטי של המצב העוקב בעל הערך היוריסטי (על סמך הפונקציה h) המינימלי מכיוון שגרף המצבים הינו עץ המצב נוכחי בפרט רחוק יותר ממצב המטרה מאשר המצבים היורשים שלו.

ג. היוריסטיקה הנדרשת לסעיף זה, זהה ליוריסטיקה מסעיף קודם פרט לשינוי עבור מצבים ש h לא מוגדרת עליהם:

```
def h1_value(State s):
    if is_goal(s)
          return 0
    h1_v = 0
    for curr_succ in succ(s).difference(problem.close):
          if ApplicableH(curr_succ):
                h_v_succ += h(curr_succ)
                if h1_v == 0 or h_v_succ < h1_v:
                      h1_v = h_v_succ
    return h1_v
```

מכיוון שאנו לא יודעים דבר על טופולוגית מרחב המצבים שלנו נרצה שהערך היוריסטי של מצב ש h לא מוגדרת עליו יחושב על סמך המצבים העוקבים שלו בעץ החיפוש. נשים לב כי מאותם נימוקים קודמים היוריסטיקה קבילה ומיודעת יותר מ - h_0.

ד. כן קיים אלגוריתם כזה: $\mathbf{IDA}^*$ אשר משתמש ביוריסטיקה הנתונה ומכיוון שהיוריסטיקה על המצב ההתחלתי היא מושלמת הוא ימצא מצב מטרה כבר באיטרציה הראשונה (החסם הראשון הוא הערך היוריסטי של המצב ההתחלתי) ומכיוון שאנו יודעים שערך זה הוא מושלם, כלומר לא קיים פתרון טוב ממנו אין טעם להמשיך לעוד איטרציה ואף ניתן לחדול את פעולת האלגוריתם כי לא ימצא מסלול טוב יותר כי הערך של המצב ההתחלתי מושלם.
הזמן ריצה שלו חסום על ידי **A*** מכיוון שהוא פועל בדיוק כמוהו רק ש **A*** ימשיך לעבוד גם כאשר הגיע למצב מטרה כי הוא לא מודע שמצא כבר מסלול הכי טוב כי לא בהכרח שמצב המטרה הוא ראשון בתור **OPEN**.